SIGNALS AND SYSTEMS STUDENTS TRAINER

Javier Montes, Javier Deza, Juan Carlos G. de Sande

Abstract

Higher education students demand fast feedback about their assignments and the opportunity to repeat them in case they do in a wrong way. Here a computer based trainer for Signals and Systems students is presented. An application, that automatically generates and assesses thousands of numerically different versions of several Signals and Systems problems have been developed. This applet guides the students to find the solution and automatically assesses and grades the students proposed solution. The students can use the application to practice in solving several types of Signals and Systems basic problems. After selecting the problem type, the student introduces a seed and the application generates a numerical version of the selected problem. Then the application presents a sequence of questions that the students must solve and the application automatically assess their answers. After solving a given problem, the students can repeat the same numerical variation of the problem by introducing the same seed to the application. In this way, they can review their solution with the help of the hints given by the application for wrong solutions. This application can also be used as an automatic assessment tool by the instructor. When the assessment is made in a controlled environment (examination classroom or laboratory) the instructor can use the same seed for all students. Otherwise, different seeds can be assigned to different students and in this way they solve different numerical variation of the proposed problem, so cheating becomes an arduous task. Given a problem type, the mathematical or conceptual difficulty of the problem can vary depending on the numerical values of the parameters of the problem. The application permits to easily select groups of seeds that yield to numerical variations with similar mathematical or conceptual difficulty. This represents an advantage over a randomised task assignment where students are asked to solve tasks with different difficulty.

Keywords: Computer based learning, automatic problem generator, automatic assessment, feedback.

1 INTRODUCTION

Computer assisted assessment and web-based assessment systems have become useful and widespread tools for educators in last years [1-7]. Many systems and platforms allowing the creation of quizzes with different types of question, adapted to the specific objective that must be reached at any step in the teaching-learning process, have been developed. The quizzes are usually created by a tailored selection of test items from a previously stored item bank. A common drawback of these systems is that preparing a large item bank is time consuming [8]. In fact, the developers of such systems emphasize that their system includes a module or a function to easily generate items and build item banks [1-3]. From the student's point of view, an important feature of the self assessment systems is the possibility of replicating the same kind of test with different data [1]. In engineering disciplines it is an easy task to repeat the same item question with different numerical data, although can be a tedious and consuming time for the instructor. In this paper, an algorithm to automatically generate numerically different exercises in the field of Signals and Systems analysis is presented. Questionnaire for assessment of the exercises solution given by the student is also proposed. Both parts are integrated in an application that can be used by the students as a tool for practising till they learn to solve studied type of problem, and by the instructor as an automatic assessment tool.

2 CONTEXT

It is a common situation in many engineering disciplines the existence of types of problems that are usually solved by following the same steps, independently of the particular parameters that define the problem. The chosen parameters determine the properties and characteristics of the particular solution. Usually the goal of the engineering problem is the analysis and the interpretation of these characteristics. A quite general example of this kind of problems is the analysis of a linear and time invariant (LTI) system both in time domain and in frequency domain and both for continuous and discrete time systems. In fact, this is one of the main objectives of Signals and Systems courses that appear in many engineering disciplines. In our case, Signal and Systems is a semesterly course programmed in the second academic year of Electrical and Electronics Engineering graduate studies at the Escuela Universitaria de Ingeniería Técnica de Telecomunicación at the Universidad Politécnica de Madrid. The course covers the following topics: introduction to signals and systems, time domain analysis of linear and time invariant systems, Fourier and Laplace analysis of continuous time signal and systems, and Fourier and Z-transform analysis of discrete time signal and systems. Time domain LTI systems analysis requires to solve the convolution of two discrete or continuous time signals. Discrete or continuous time LTI systems can be described by a linear constant coefficient difference or differential equation (LCCDE). Although, from a mathematical point of view, convolution and LCCDE solvers are implemented in many mathematical packages [9], it is still interesting that the students learn to solve simple cases without the aid of such mathematical packages. Random generation of the coefficients that define the LCCDE or the signals to be convolved could be an easy way to automatically generate thousands of problems. This way to generate numerical variations of a problem could easily yields to nonsense numerical solutions. For example, real time systems are always causal systems, and they have frequency response only if they are stable systems [10]. On the other hand, the mathematical solutions for those problems are easy to obtain (without the help of software) only for certain sets of parameters, so it is necessary to find the way to automatically generate these sets of parameters.

Algorithms to automatically generate two different signals to be convolved or the coefficients that define a LCCDE (continuous time case) have been proposed [11-12]. Here, the discrete time LCCDE case is also included. Generated coefficients are chosen in such a way that the following conditions are satisfied:

- The signals (convolution case) should be simple enough to be easily convolved
- The system should be stable (in order to have frequency response) and causal (in order to the system has physical meaning, as a mechanical damper or a passive electric circuit). This is satisfied if the poles of the system are inside the unit circle in the z-plane for discrete time systems or if the poles are in the negative real part s-plane [10].
- All the coefficients should be not too high integers (up to 2 digits). Then the required calculations will be quite easy. The purpose is that the student will learn the mechanical steps he/she must follow to obtain the particular solution.
- A large enough set of different problems should be generated. The purpose of this condition is that each student that attends a particular course will have enough numerically different exercises to be solved.
- In order to generate a particular set of coefficients vectors, an integer number is enough as input to the algorithm.

3 DESCRIPTION OF THE APPLICATION

An application has been developed to implement the automatic generation of thousands of variations of several classes of problems and its automatic assessment. The numerical variations of the selected problem are generated in a controlled way so it permits a personalized task assignment. An automatic correction and evaluation of solution given by the students attending to qualitative and quantitative aspects is also presented.

We have chosen to develop this application in Java [13] for some of its main advantages:

- Object Oriented Programming. Java creates modular and flexible programs. One of the main requirements of the developed application is to be modular, so it is easy to add different types of exercises.
- Platform independence. A Java application runs on virtually any platform and operating system. The main objective of the application is that students can use it as a learning tool by the repetition of an exercise. The students should have the possibility of using the application at any place and whenever they want. Java is used to create applications called Applets. The applets run inside the Web browsers, so these applications can be used from any computer with internet connection. The applets can be incorporated in SCORM [14] packages that can be used in any LMS like Moodle [5].

Fig. 1 shows a block diagram of the developed application. At the beginning, the user should select a problem type. The application selects a given applet that generates a problem variation and guides the users to solve it. The users are required to answer a series of questions (its number can vary depending on the problem type and, for a given problem type, on the user answers) that are automatically assessed. A score manager finally shows a global mark for the trial. Fig. 1 shows the three types of problems developed until now, but more applets could be added in a modular way.

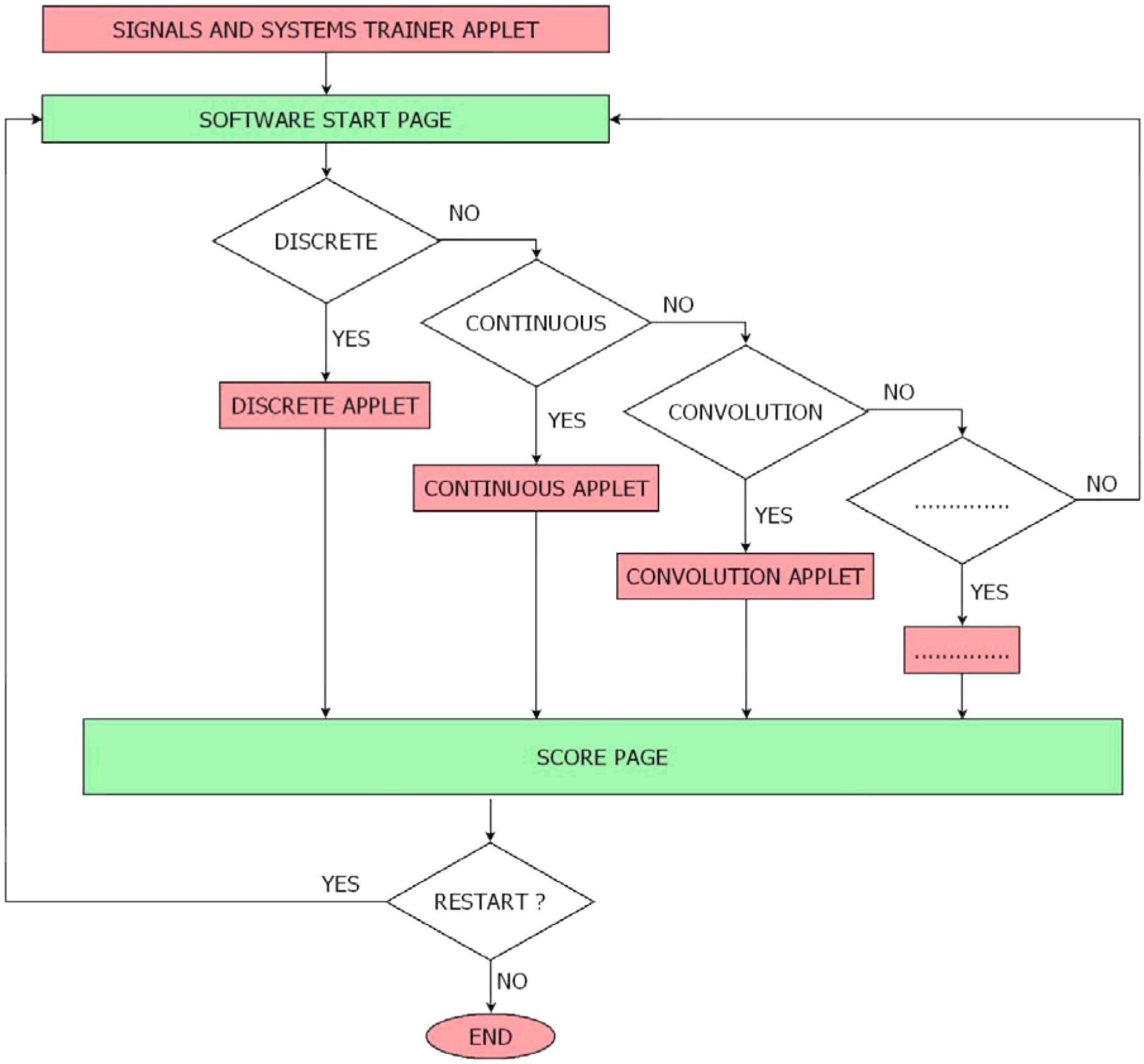

Figure 1: Block diagram of the signals and system trainer application

Discrete time, continuous time and convolution applets have a common structure shown in Fig. 2. This structure should be followed when creating a new applet with a new problem type. To generate a new numerical version of the selected problem, the user is required to introduce an arbitrary integer number. Depending on the problem type, between 2000 and 25000 numerical variations of the problem can be generated [11, 12]. The use of a student selected seed instead of a random seed has been implemented because in this way the student can repeat the same numerical variation of a problem as many times as he/she wants. Then a new screen with the problem wording and the first questions appears. After solving and answering these questions, user should click on the “Assess” button. Then a solution manager block obtains the actual solutions for all questions and compares them with the user’s answers. After assessing each questions page, a feedback message is shown for each question. The feedback message includes “Ok” or “Wrong” alert, the right solution and, in some cases, a hint for wrong answers. User should click on the “Next” button to continue solving the problem. This process is repeated till the end of the problem. Then, the student can exit the application or go back to the beginning in order to solve a new problem or in order to repeat the same one.

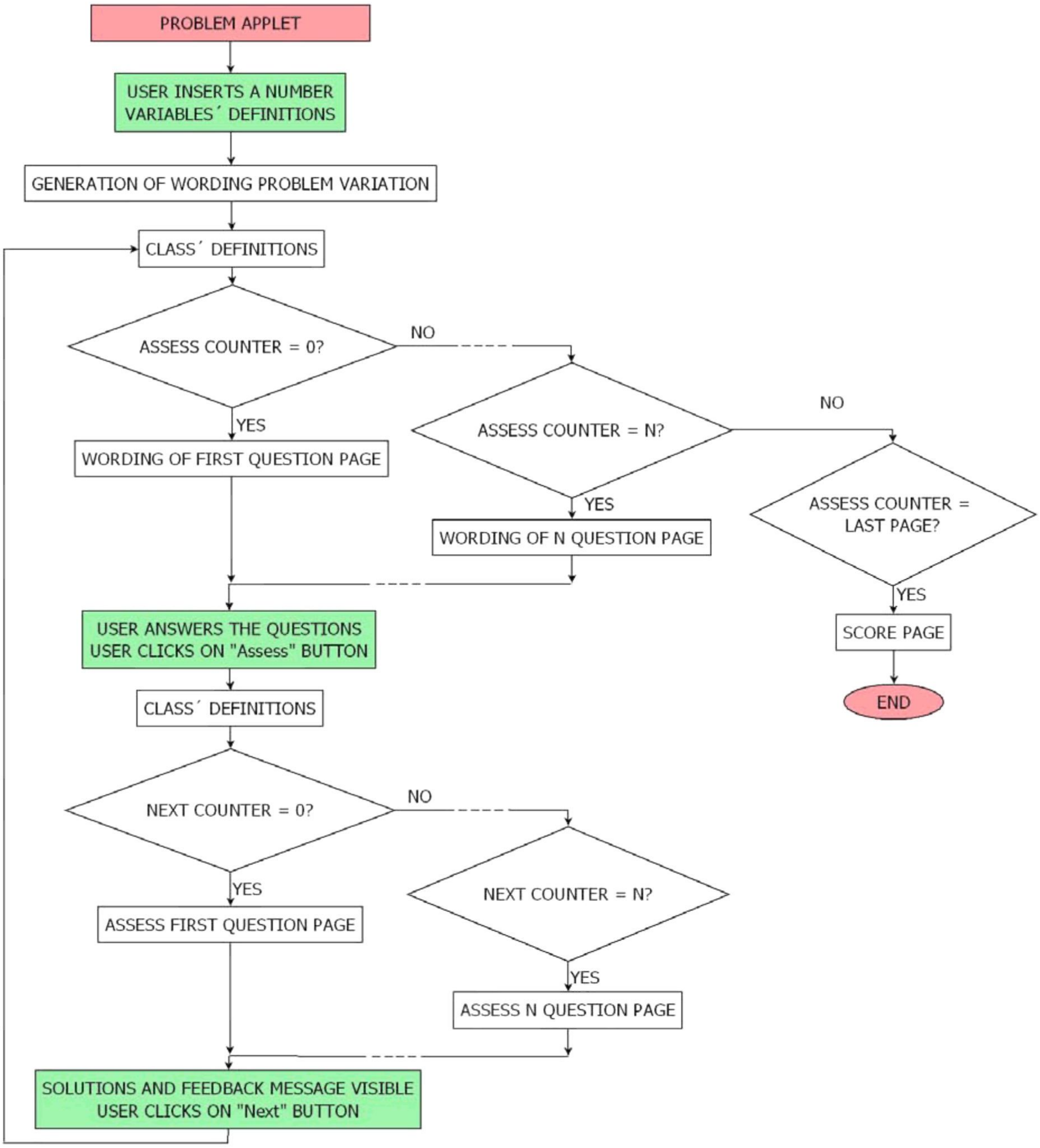

Figure 2: Block diagram of the continuous time LTI system analysis applet. The same scheme is followed for all the problem types

The assessment manager block is described in Fig. 3. First of all, this assessment manager uses the seed (the integer number introduced by the student after selecting a problem type) to obtain the actual solution of all questions for the present numerical variation of the selected problem. When user clicks on “Assess” button of any page, the assessment manager checks that user have answered all questions, if it is not the case, the user is required to fill in all questions. When the user have input an answer for all questions in a page, the assessment manager compares them with the actual solutions and assigns a score (0 or the selected weight) for each question.

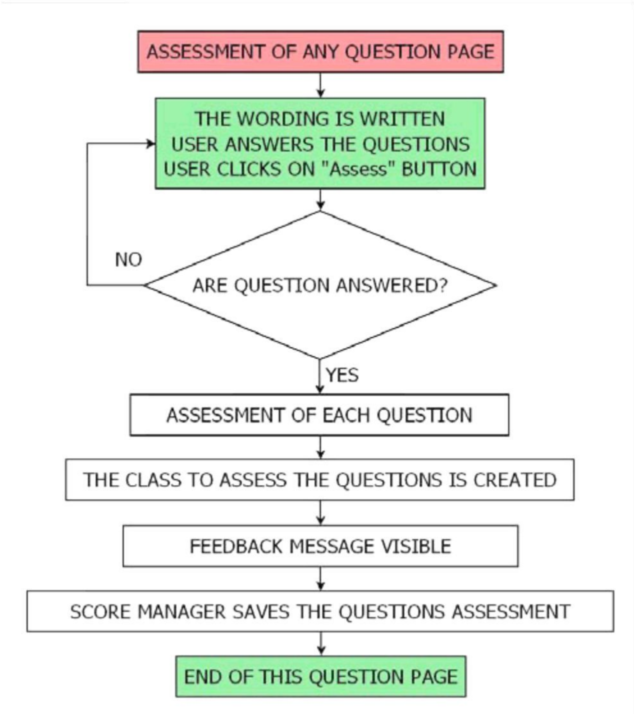

Figure 3: Block diagram of the continuous time LTI system analysis applet. The same scheme is followed for all the problem types

4 EXAMPLES

The automatic generation of parameters that define exercises where continuous time LTI systems are analyzed in transform domains and the automatic generation of two continuous time or discrete time signals to be convolved (time domain LTI system analysis) has been previously described [11-12]. In the next subsection, the automatic generation of numerical values that define a discrete time LTI system is introduced.

4.1 Discrete time, linear and time invariant systems analysis

A typical exercise that can be found in many Signals and Systems textbooks [10] is the following:

"The input x[n] and the output y[n] of a causal LTI system are related by the difference equation:

$$a_0 y[n] + a_1 y[n-1] + a_2 y[n-2] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2]. \qquad (1)$$

Find the system function and draw its pole-zero diagram.

b) Based on the pole-zero diagram, sketch the impulse response indicating its qualitative behaviour.

c) Find the impulse response of the system. Is the duration of this impulse response finite or infinite?

d) Find the frequency response of the system

e) Etc…"

The exercise solution will be quantitatively and qualitatively different depending on the particular set of coefficients. In the following, a method to generate the $a=[a_2, a_1, a_0]$ and $b=[b_2, b_1, b_0]$ coefficients from a given integer number is proposed.

4.1.1 Algorithm to generate a and b vectors

The method is based in the repeated application of modulo operation (which gives the remainder of the division of two numbers) on a given input number. This modulo operation is applied with different divisors in order to get a set of low integer values (from 0 to 10) that will be used to generate the constant coefficients of the difference equation. The idea can be easily applied to any kind of parameter dependent problems.

The starting point of the algorithm is an input integer number (N_0 in the following). The second step is to calculate the N_0 modulo n values. In the present form of the algorithm the following values m_2=N_0 modulo 2, m_3=N_0 modulo 3, m_5=N_0 modulo 5, m_7=N_0 modulo 7, and m_{11}=N_0 modulo 11, are calculated. Additional values of m_n where n are the following prime numbers can be added if we want to generate a larger set of *a* and *b* vectors, and then a larger set of exercises.

The third step is to decide which kind of system, depending on the duration of impulse response, will be generated. A finite impulse response (FIR) will be generated when m_2=0, and an infinite impulse response (IIR) be generated when m_2=1.

The fourth step is to build up *a* and *b* vectors using the remaining m_n values. Only first and second order systems are considered in order to reduce the necessary mathematical work to solve the exercise. First order filters are selected when m_3=0, and second order filters when m_3=1 or 2.

The selected values of *a* and *b* vectors for first order (m_3=0) FIR filters (m_2=0) are:

$$\begin{aligned} a &= 4(1+m_5) \\ b &= \left[4+m_{11}, \quad (4+m_7)(-1)^{m_5}\right] \end{aligned} \tag{2}$$

and for the case of second order ($m_3 \neq 0$) FIR filters, the selected values of *a* and *b* vectors are:

$$\begin{aligned} a &= 4(1+m_5) \\ b &= \left[1+m_7, \quad (m_{11}-m_7-2)(-1)^{m_3}, \quad 1+m_5\right] \end{aligned} \tag{3}$$

Following these rules, $5\times7\times11=385$ first order and $2\times5\times7\times11=770$ second order FIR filters are generated with integer coefficients in the −8 to 20 range.

In the case of first order (m_3=0) IIR filters (m_2=1), the selected values of *a* and *b* vectors are:

$$\begin{aligned} a &= \left[3(3+m_5), \quad 6(-1)^{m_5}\right] \\ b &= \left[6+m_7, \quad (-4-m_{11})\times(-1)^{m_5}\right] \end{aligned} \tag{4}$$

Finally, second order ($m_3 \neq 0$) IIR filters are generated by selecting the following values:

$$\begin{aligned} a &= \left[2(1+m_3)/m_3 \quad 2(2-m_5)/m_3 \quad 1\right] \\ b &= \begin{cases} \left[1+m_7 \quad (m_{11}-m_7-2)(-1)^{m_3} \quad -1-m_5\right] & \text{if } m_5 = 2 \\ \left[1+m_7 \quad (2+m_7-m11)(-1)^{m_3} \quad 1+m_5\right] & \text{if } m_5 \neq 2. \end{cases} \end{aligned} \tag{5}$$

It is observed that the generated elements of *a* and *b* vectors are integer numbers in the −14 to 21 range. Following these rules, there are 3 integers in every 2310 consecutive integer numbers that yields to *a* and *b* vectors related by $a = k \times b$, where *k* is a constant (an integer for the chosen *a* and *b* values) which yields to a proportional relation between the input and the output of the system. There is a fourth integer in every 2310 consecutive integer numbers (for which m_3=1) that yields to *a* and *b* vectors such that a pole of the system coincides with a zero of the system, then, the system is reduced to a first order system. Then, $5\times7\times11+1=386$ first order IIR filters, and $2\times5\times7\times11-4=766$ second order IIR filters are generated. Additionally, three zero order LTI systems are also obtained.

With the proposed method, 2310 different pairs of *a* and *b* vectors are generated. They satisfy the criteria C.1 to C.4, and define numerically different exercises. This number can be easily increased by using m_{13}, m_{17}, etc.

Fig. 4 shows an example of the wording of this exercise generated by the application and the first questions page.

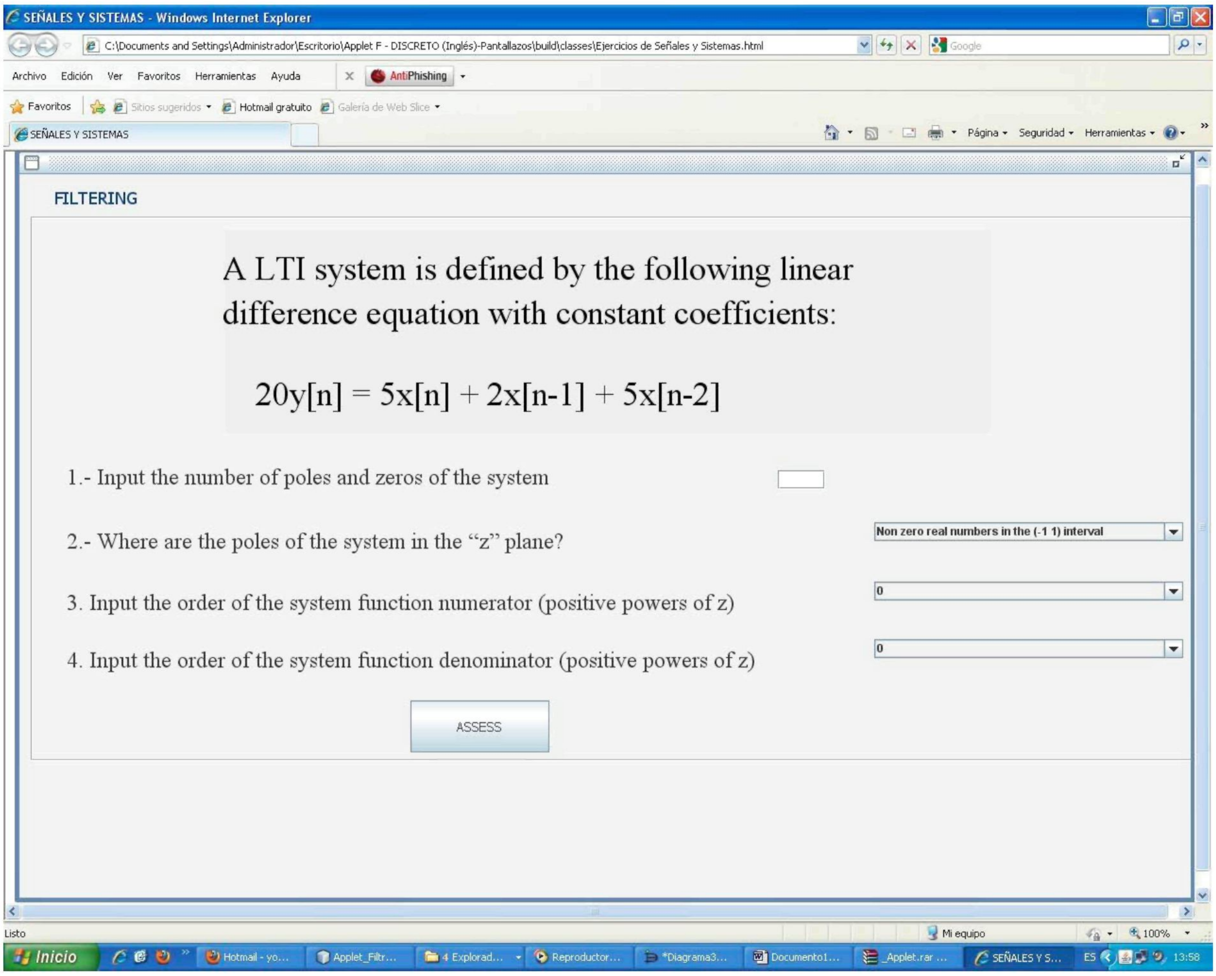

Figure 4: Example of the discrete time LTI system analysis applet

5 CONCLUSIONS

An application, that automatically generates and assesses thousands of numerically different versions of several Signals and Systems problems have been developed. This applet guides the students to find the solution and automatically assesses and grades the student's proposed solution. Students can use the application to practice in solving several types of Signals and Systems basic problems. After selecting the problem type, the user introduces a seed and the application generates a numerical version of the selected problem. Then, the application presents a sequence of questions that the user must solve and the application automatically assesses the student's answers. After solving a given problem, the students can repeat the same numerical variation of the problem by introducing the same seed to the application. In this way, they can review their solution with the help of the hints given by the application for wrong solutions.

REFERENCES

[1] Bruce-Lockhart, M. Crescenzi, P. Norvell, T. 2009. Integrating test generation functionality into the Teaching Machine environment. *Electronic Notes in Theoretical Computer Science*, 224, pp.115–124

[2] He, Q. & Tymms, P. 2005. A computer-assisted test design and diagnosis system for use by classroom teachers. *Journal of Computer Assisted Learning*, 21 (6), pp. 419-429.

[3] Wang, T. H. Wang, K. H. Wang, W. L. Huang, S. C. & Chen, S. Y. 2004. Web-based Assessment and Test Analyses (WATA) system: development and evaluation. *Journal of Computer Assisted Learning*, 20 (1), pp. 59-71.

[4] Angus, S.D. & Watson, J. 2009. Does regular online testing enhance student learning in the numerical sciences? Robust evidence from a large data set. *British Journal of Educational Technology*, 40 (2), pp. 255-272.

[5] Moodle.org: open-source community-based tools for learning, http://www.moodle.org, [Accessed 10 May 2011].

[6] Shih, C.C. Chiang, D.A. Lai, S.W. & Hu, Y.W. 2009. Applying hybrid data mining techniques to web-based self-assessment system of Study and Learning Strategies Inventory. *Expert Systems with Applications*, 36, pp.5523–5532.

[7] Wang, T.H. 2007. What strategies are effective for formative assessment in an e-learning environment? *Journal of Computer Assisted Learning*, 23 (3), pp. 171-186.

[8] Huang, Y. M. Lin, Y.T. & Cheng, S.C. 2009. An adaptive testing system for supporting versatile educational assessment. *Computers & Education*, 52, pp. 53–67.

[9] Matlab: http://www.mathworks.com/products/matlab/, [Accessed 10 May 2011].

[10] Oppenheim, A.V. Willsky, A. S. & Nawab, S. H. 1997. *Signals and Systems*. Upper Saddle River, New Jersey: *Prentice-Hall*.

[11] J.C.G. de Sande. Automatic generation of Signals and Systems exercises. Proceedings of ICERI2010 conference. 15-17 November 2010 Madrid, Spain.

[12] A. Osorio and J.C.G. de Sande. Computer based trainer for Signals and Systems undergraduate students. Proceedings of ICERI2010 conference. 15-17 November 2010 Madrid, Spain.

[13] Ian F. Darwin. 2004. *JAVA Cookbook*. O´Reilly Media.

[14] SCORM: http://scorm.com/scorm-solved/scorm-engine/ [Accessed 18 May 2011].